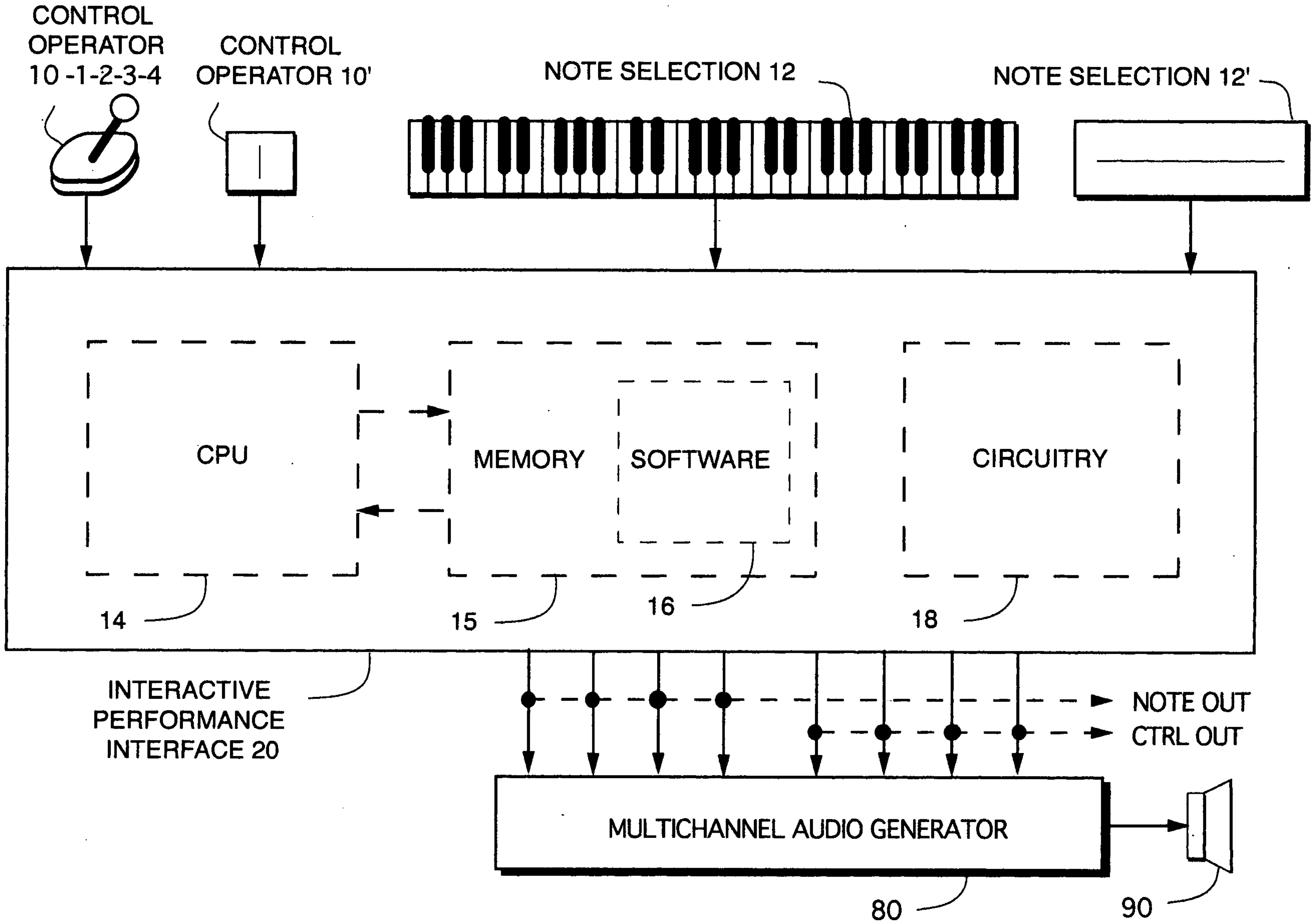

CONTROL
OPERATOR
10 -1-2-3-4
CONTROL
OPERATOR 10'
NOTE SELECTION 12
NOTE SELECTION 12'
CPU
MEMORY
SOFTWARE
CIRCUITRY
14
15
16
18
INTERACTIVE
PERFORMANCE
INTERFACE 20
NOTE OUT
CTRL OUT
MULTICHANNEL AUDIO GENERATOR
80
90

FIGURE 1

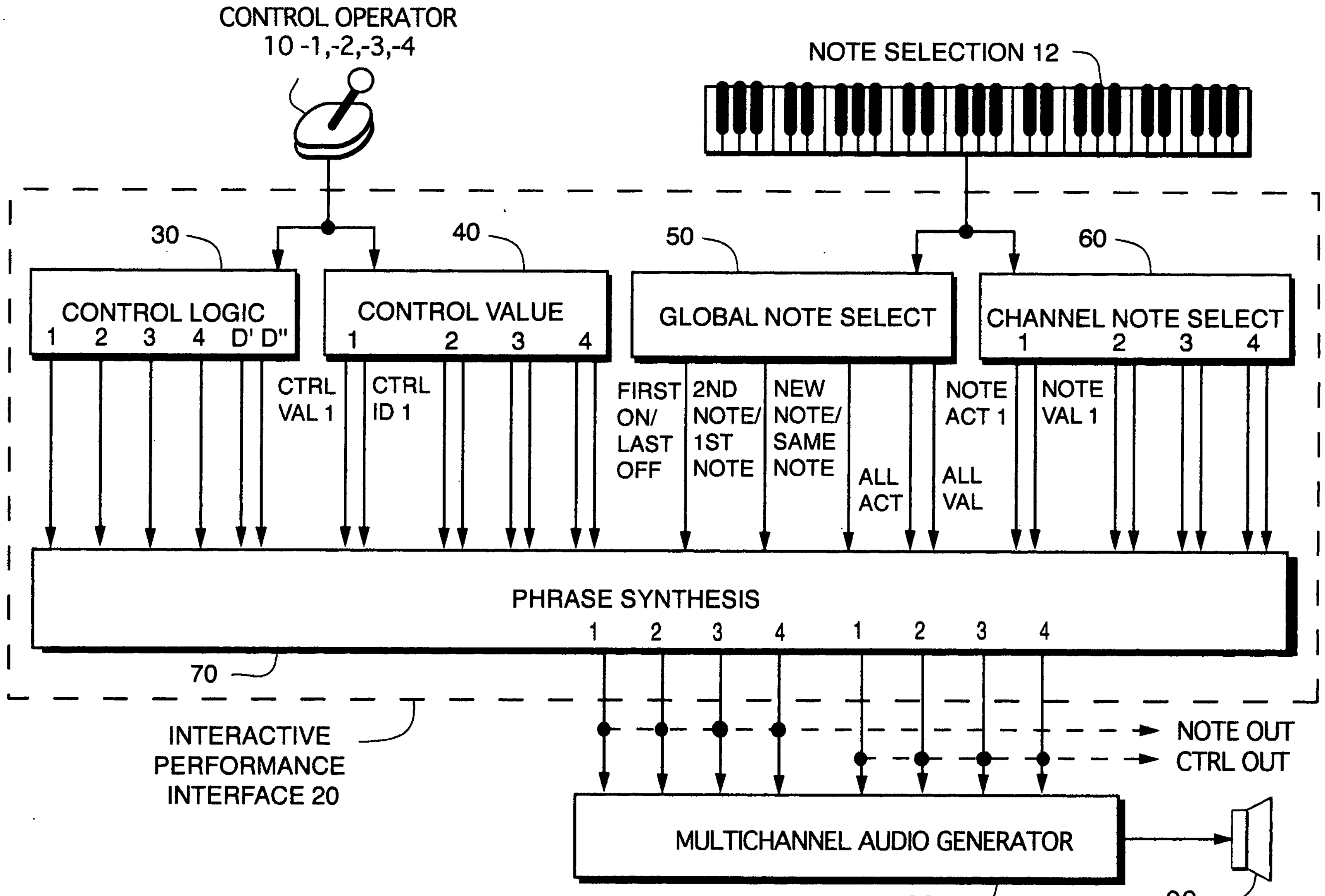

CONTROL OPERATOR
10 -1,-2,-3,-4
NOTE SELECTION 12
30
40
50
60
CONTROL LOGIC
1 2 3 4 D' D''
CONTROL VALUE
1 2 3 4
GLOBAL NOTE SELECT
CHANNEL NOTE SELECT
1 2 3 4
CTRL VAL 1
CTRL ID 1
FIRST ON/ LAST OFF
2ND NOTE/ 1ST NOTE
NEW NOTE/ SAME NOTE
ALL ACT
NOTE ACT 1
ALL VAL
NOTE VAL 1
PHRASE SYNTHESIS
1 2 3 4 1 2 3 4
70
INTERACTIVE PERFORMANCE INTERFACE 20
NOTE OUT
CTRL OUT
MULTICHANNEL AUDIO GENERATOR
80
90

FIGURE 2

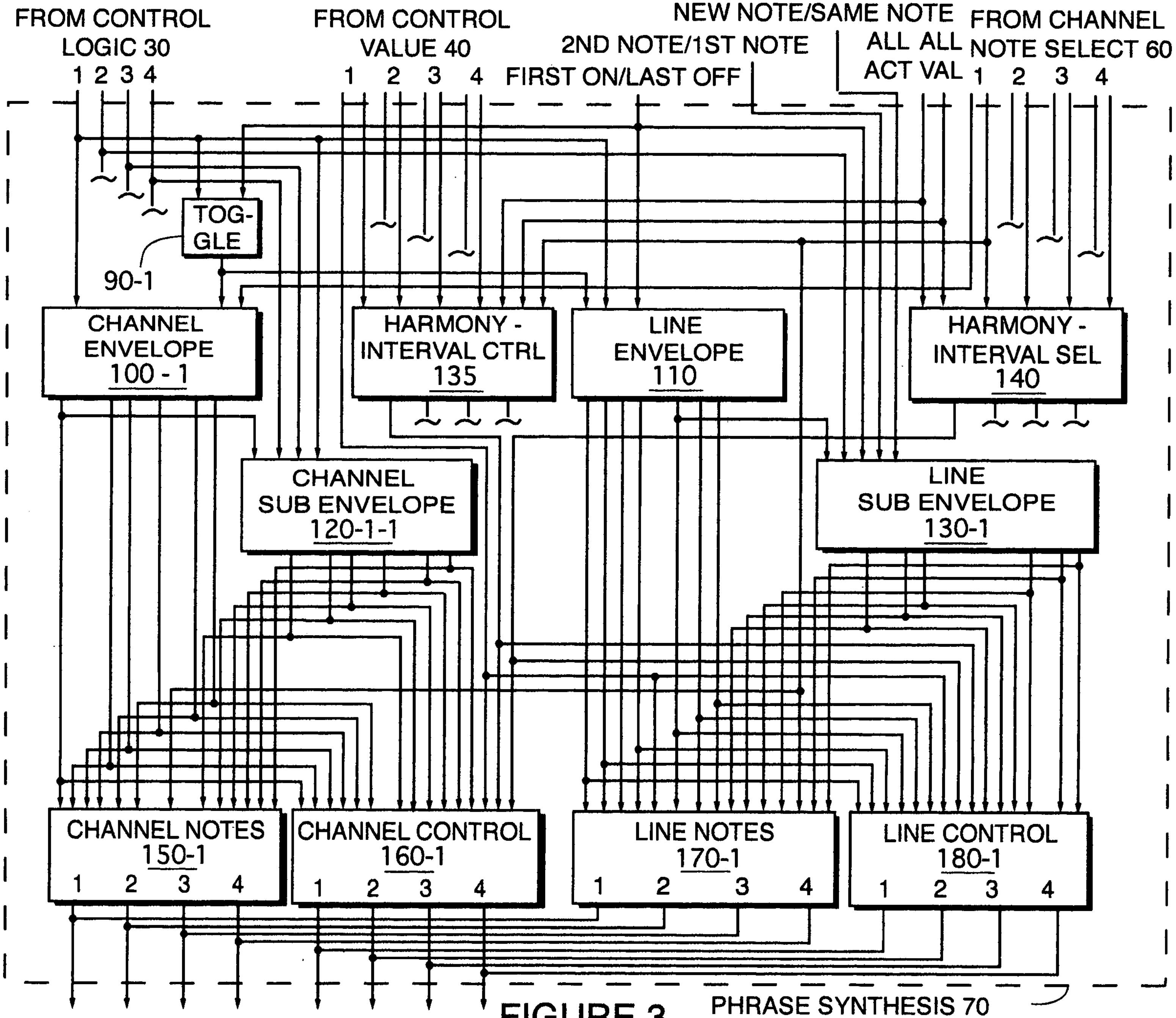
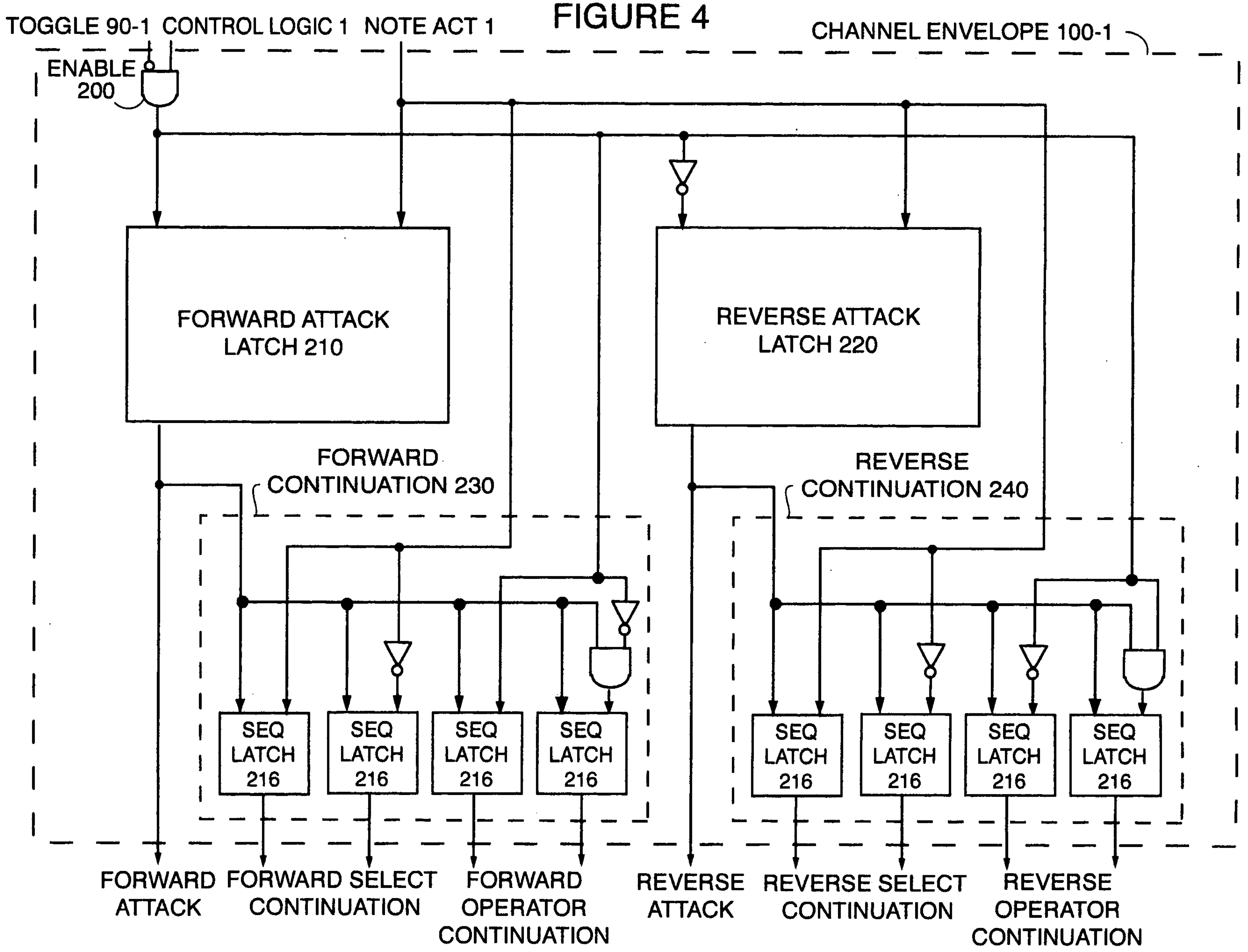

FROM CONTROL LOGIC 30
1 2 3 4
FROM CONTROL VALUE 40
1 2 3 4
NEW NOTE/SAME NOTE
2ND NOTE/1ST NOTE
FIRST ON/LAST OFF
ALL ACT
ALL VAL
FROM CHANNEL NOTE SELECT 60
1 2 3 4
TOG-GLE
90-1
CHANNEL ENVELOPE 100 - 1
HARMONY - INTERVAL CTRL 135
LINE ENVELOPE 110
HARMONY - INTERVAL SEL 140
CHANNEL SUB ENVELOPE 120-1-1
LINE SUB ENVELOPE 130-1
CHANNEL NOTES 150-1
1 2 3 4
CHANNEL CONTROL 160-1
1 2 3 4
LINE NOTES 170-1
1 2 3 4
LINE CONTROL 180-1
1 2 3 4
PHRASE SYNTHESIS 70

FIGURE 3

FIGURE 4



TOGGLE 90-1
CONTROL LOGIC 1
NOTE ACT 1
CHANNEL ENVELOPE 100-1
ENABLE
200
FORWARD ATTACK LATCH 210
REVERSE ATTACK LATCH 220
FORWARD CONTINUATION 230
REVERSE CONTINUATION 240
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
SEQ LATCH 216
FORWARD ATTACK
FORWARD SELECT CONTINUATION
FORWARD OPERATOR CONTINUATION
REVERSE ATTACK
REVERSE SELECT CONTINUATION
REVERSE OPERATOR CONTINUATION

FIGURE 5A

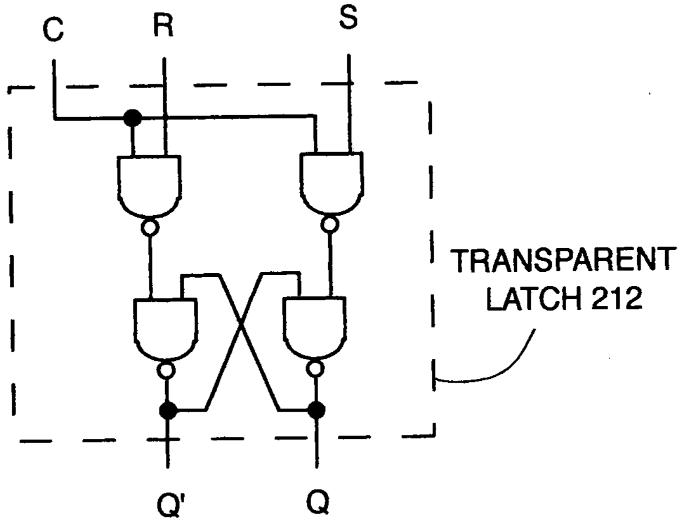


FIGURE 5B

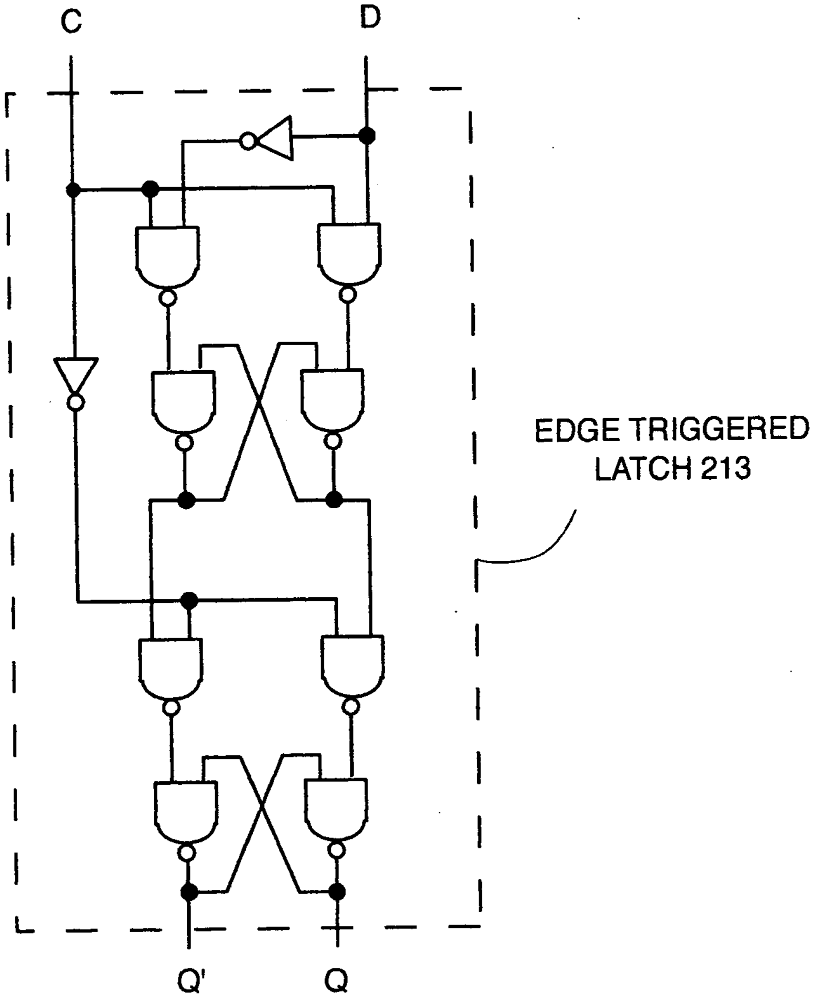


FIGURE 6A

AND GATE 211

```
Public Function AndGate(ByVal Left As Boolean, ByVal Right As Boolean) As Boolean

 If Left And Right Then
  AndGate = True
 End If

End Function
```

FIGURE 6B

TRANSPARENT
LATCH 212

```
Public Function TransLatch(In1 As Boolean, In2 As Boolean) As Boolean
 Static Trans As Boolean

 Trans = Not AndGate(Not (AndGate(In1, Not (In2))), _
         Not (AndGate(Not (AndGate(In1, In2)), Trans)))
 TransLatch = Trans

End Function
```

FIGURE 6C

EDGE TRIGGERED
LATCH 213

```
Public Function Latch(In1 As Boolean, In2 As Boolean) As Boolean
 Static Latched, Transed As Boolean

 Latched = AndGate(Not (AndGate(Not (In1), Transed)), _
         Not (AndGate(Not (AndGate(Not (In1), _
         Not (AndGate(Not (AndGate(In1, In2)), Transed)))), Not (Latched))))
 Transed = TransLatch(In1, In2)
 Latch = Latched

End Function
```

FIGURE 7A

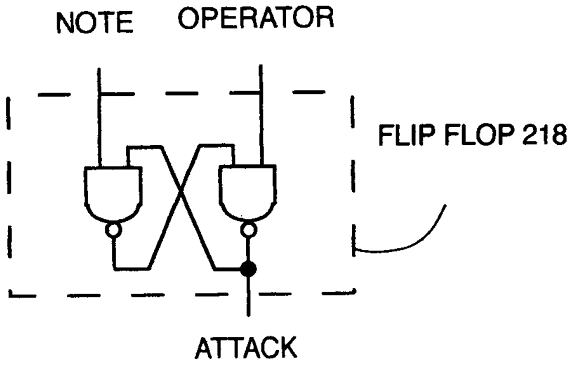


FIGURE 7B

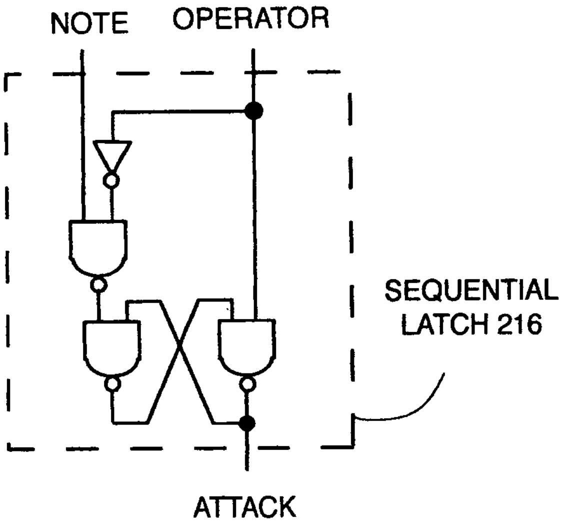


FIGURE 7C

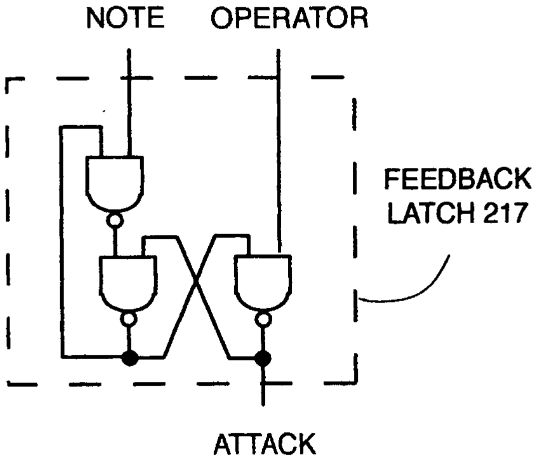


FIGURE 8A

FLIP FLOP 218

```
Public Attack As Boolean

Function FlipFlop(Note, Op as Boolean)

   If Not Note and Op and Attack Then Attack = False
   If Not Op and Not Attack then Attack = True

   Debug.Print Attack
End Function
```

FIGURE 8B

SEQUENTIAL LATCH 216

```
Public Attack As Boolean

Function SeqLatch(Note, Op as Boolean)
Static Gate as Boolean

   If Gate And Op Then Attack = True
   If Not Op Then Attack = False
   If Note And Not Op Then Gate = True
   If Not Note Then Gate = False

   Debug.Print Attack
End Function
```

FIGURE 9A

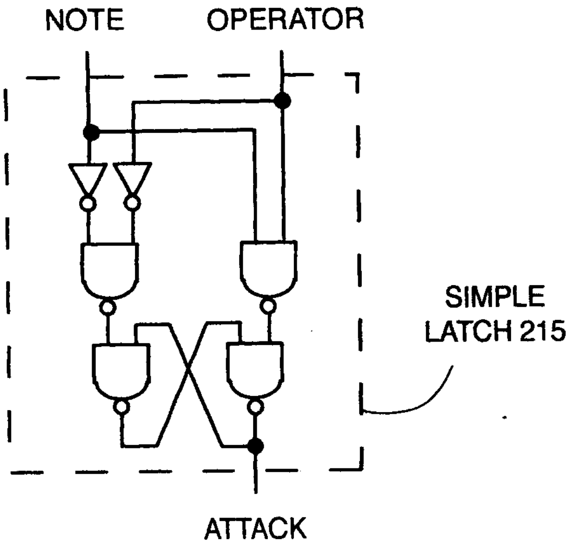


FIGURE 9B

SIMPLE LATCH 215

```
Public Function AttRel1(Note As Boolean, Op As Boolean) As Boolean
Static AttackLeft as Boolean

    If Note And Op And Not AttackLeft Then
        AttackLeft = True
    End If

    If Not Note And Not Op And AttackLeft Then
        AttackLeft = False
    End If

    AttRel1 = AttackLeft

End Function
```

FIGURE 10A

```
Public AttackLeft, AttackRight As Boolean

Public Function AttRel1(Note As Boolean, Op As Boolean) As Boolean
Static AttRel1 as Boolean

      If Note And Not Op Then
         AttackLeft = True
      End If

      If Not Note And Not Op Then
         AttackLeft = False
      End If

      AttRel1 = AttackLeft

End Function
```

FIGURE 10B

```
Public AttackLeft, AttackRight As Boolean

Function AttRel2(Note As Boolean, Op As Boolean)

      If Note And Not Op And Not AttackRight Then
         AttackLeft = True
      End If

      If Note And Op And Not AttackLeft Then
         AttackRight = True
      End If

      If Not Note And Op Then
         AttackRight = False
      End If

      If Not Note And Not Op Then
         AttackLeft = False
      End If

      Debug.Print AttackLeft
      Debug.Print AttackRight

End Function
```

FIGURE 11A

```
Public AttackLeft, as Boolean

Function AttRel8(Note As Boolean, Op As Boolean)
Static Gate1, Gate2 as Boolean

        If Note And Not Op And Gate1 Then
           AttackLeft = True
        End If
        If Not Note And Op And Not Gate1 Then
           AttackLeft = False
        End If
        If Not Note And Not Op And Not AttackLeft Then
           Gate1 = True
        End If
        If Not Note And Not Op And AttackLeft Then
           Gate1 = False
        End If
        Debug.Print AttackLeft

End Function
```

Attack Latch 210

FIGURE 11B

```
Public AttackLeft, AttackRight as Boolean

Function AttRel5(Note As Boolean, Op As Boolean)
Static Gate1, Gate2 as Boolean

        If Note And Not Op And Not AttackRight Then
           AttackLeft = True
           Gate1 = False
        End If
        If Note And Op And Not AttackLeft Then
           AttackRight = True
           Gate2 = False
        End If
        If Not Note And Op Then
           Gate2 = True
        End If
        If Not Note And Not Op Then
           Gate1 = True
        End If
        If Not Note And Not Op And Gate2 Then
           AttackRight = False
        End If
        If Not Note And Op And Gate1 Then
           AttackLeft = False
        End If
        Debug.Print AttackLeft
        Debug.Print AttackRight
 End Function
```

FIGURE 12

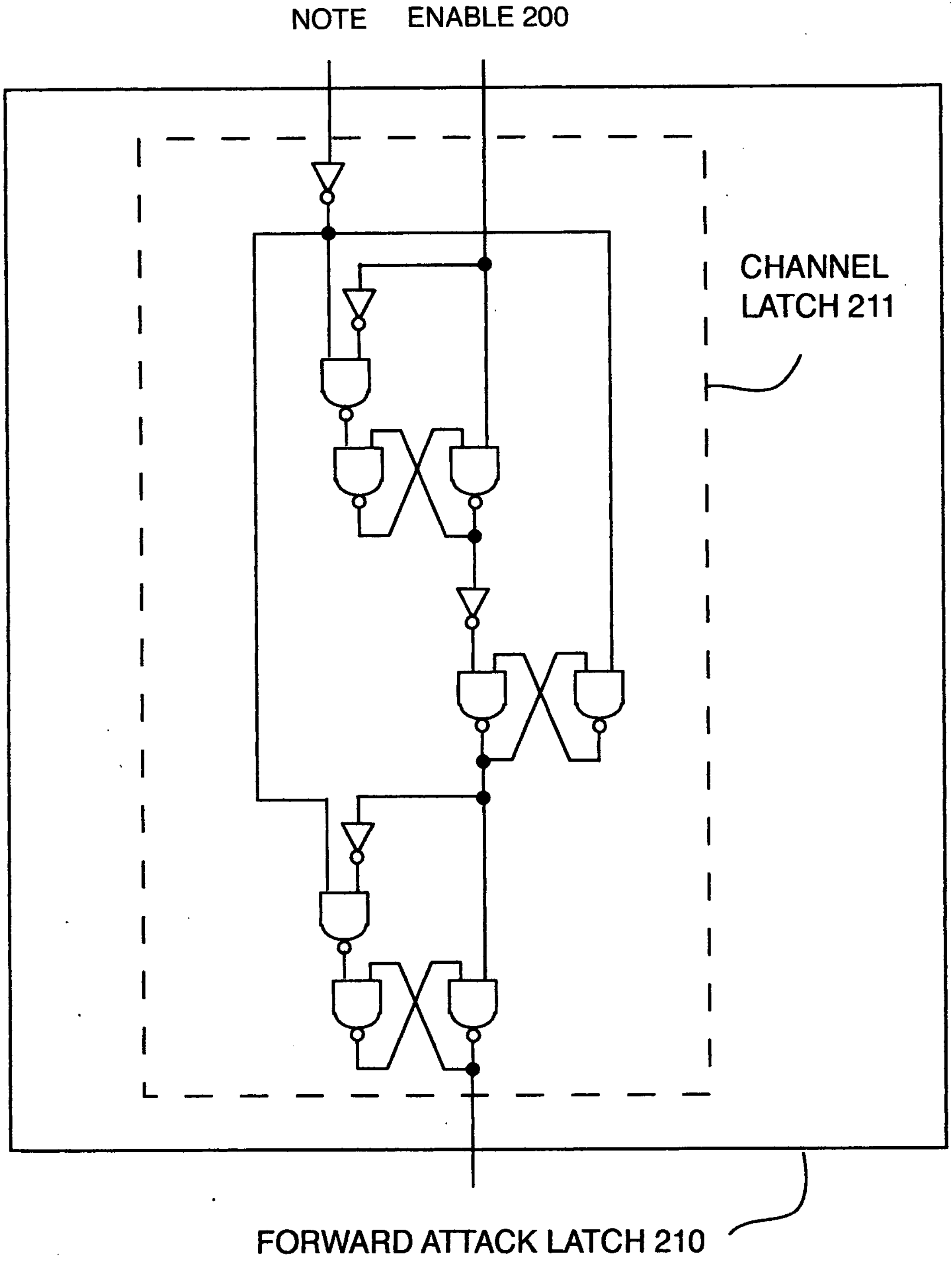


FIGURE 13

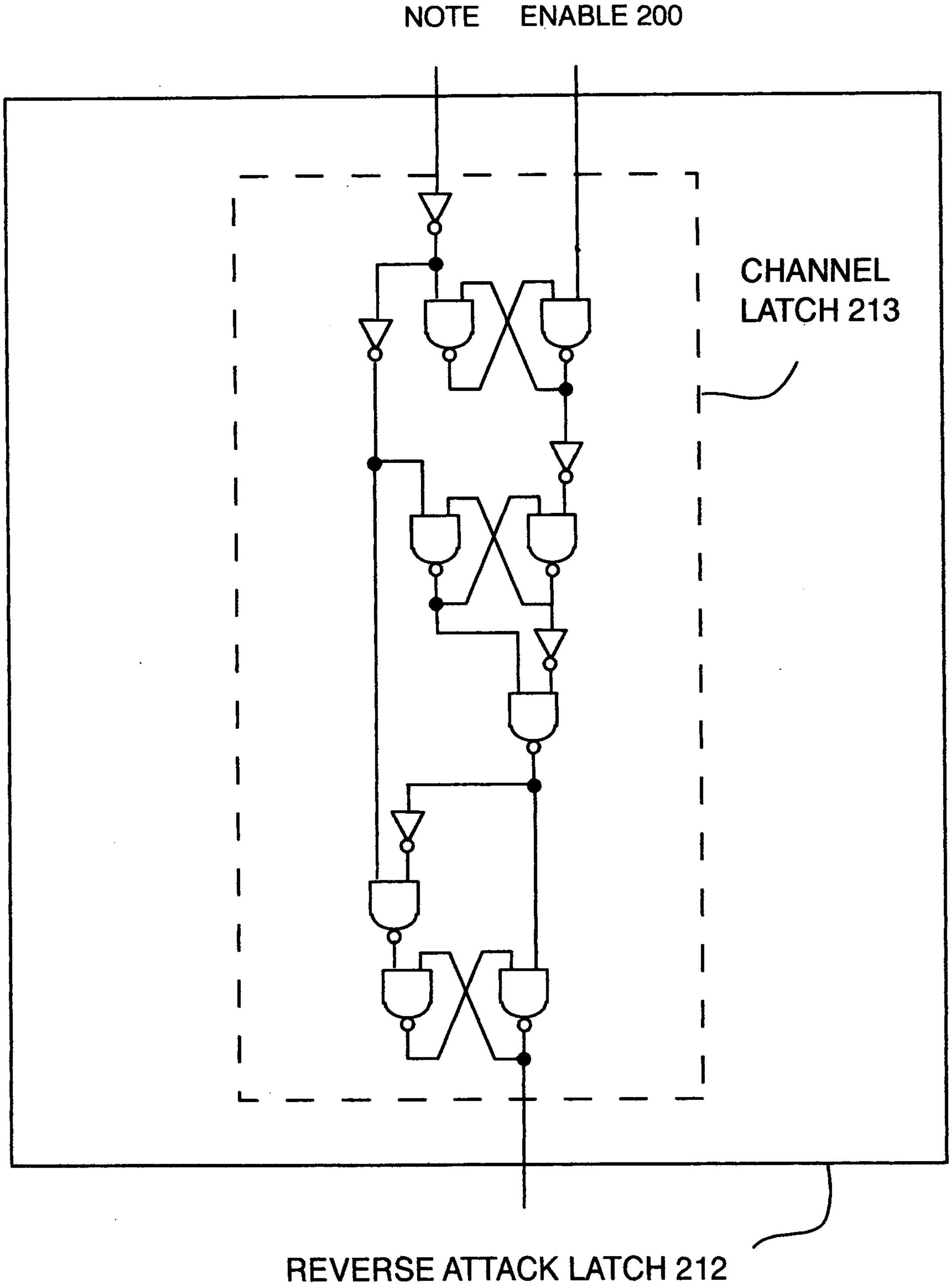


FIGURE 14

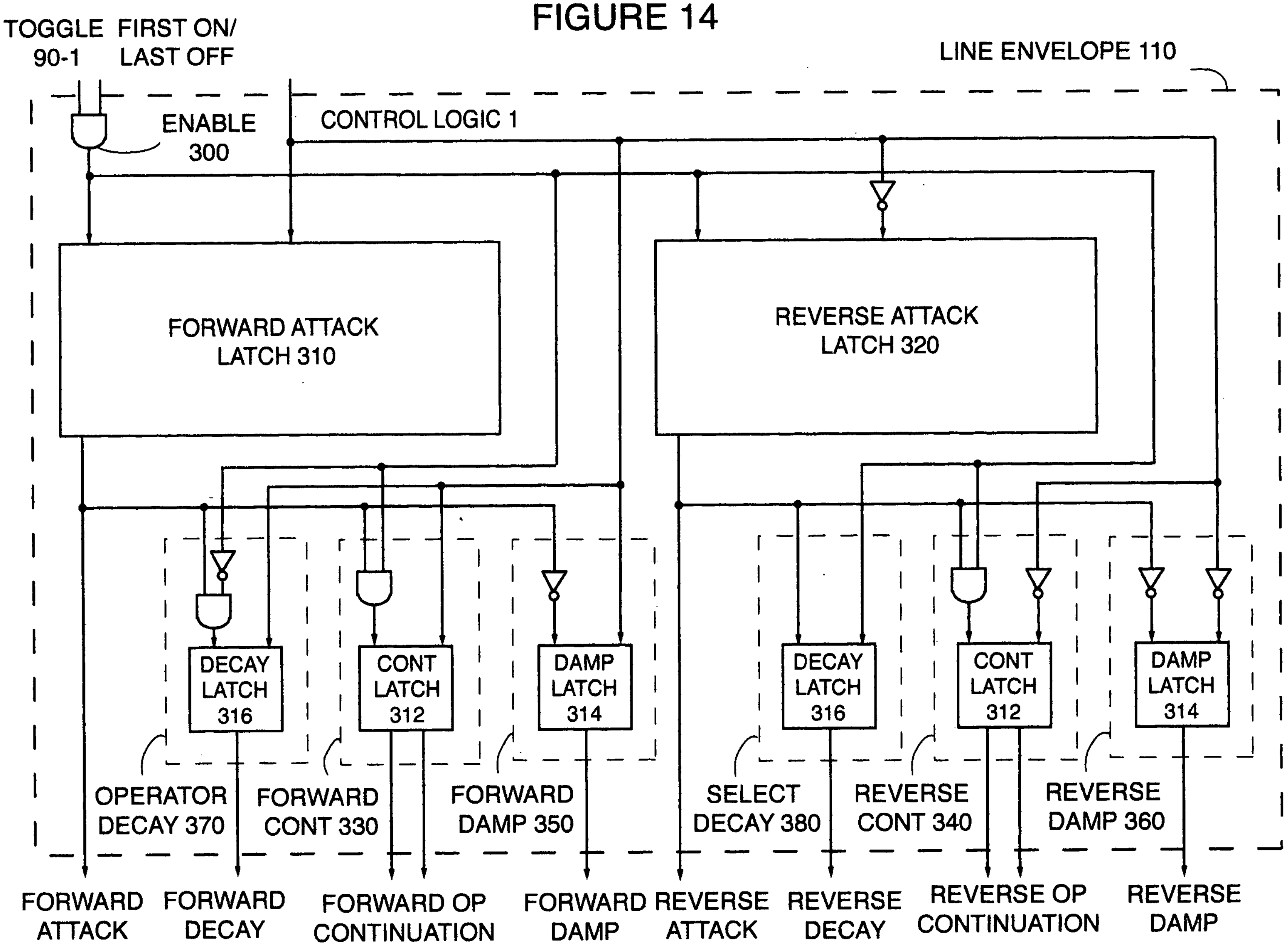

TOGGLE 90-1
FIRST ON/ LAST OFF
LINE ENVELOPE 110
ENABLE 300
CONTROL LOGIC 1
FORWARD ATTACK LATCH 310
REVERSE ATTACK LATCH 320
DECAY LATCH 316
CONT LATCH 312
DAMP LATCH 314
DECAY LATCH 316
CONT LATCH 312
DAMP LATCH 314
OPERATOR DECAY 370
FORWARD CONT 330
FORWARD DAMP 350
SELECT DECAY 380
REVERSE CONT 340
REVERSE DAMP 360
FORWARD ATTACK
FORWARD DECAY
FORWARD OP CONTINUATION
FORWARD DAMP
REVERSE ATTACK
REVERSE DECAY
REVERSE OP CONTINUATION
REVERSE DAMP

FIGURE 15

```
Public AttackLeft, AttackRight As Boolean

Function AttRel4(Note As Boolean, Op As Boolean)

If Note And Op And Not AttackRight Then
   AttackLeft = True
End If

If Note And Not Op And Not AttackLeft Then
   AttackRight = True
End If

If Not Note And Not Op Then
   AttackLeft = False
End If

If Not Note And Op Then
   AttackRight = False
End If

Debug.Print Attack Left
Debug.Print Attack Right

End Function
```

FIGURE 16A

FORWARD ATTACK
LATCH 310

```
Public AttackLeft as Boolean

Public Function AttRel7(Note As Boolean, Op As Boolean) As Boolean
Static Gate1, Gate2 as Boolean

    If Note And Op And Gate2 Then
        AttackLeft = True
    End If
    If Not Note And Not Op And Not Gate2 Then
        AttackLeft = False
    End If
    If Not Note And Op Then
        Gate2 = False
    End If
    If Note And Not Op Then
        Gate2 = True
    End If
    Debug.Print AttackLeft

End Function
```

FIGURE 16B

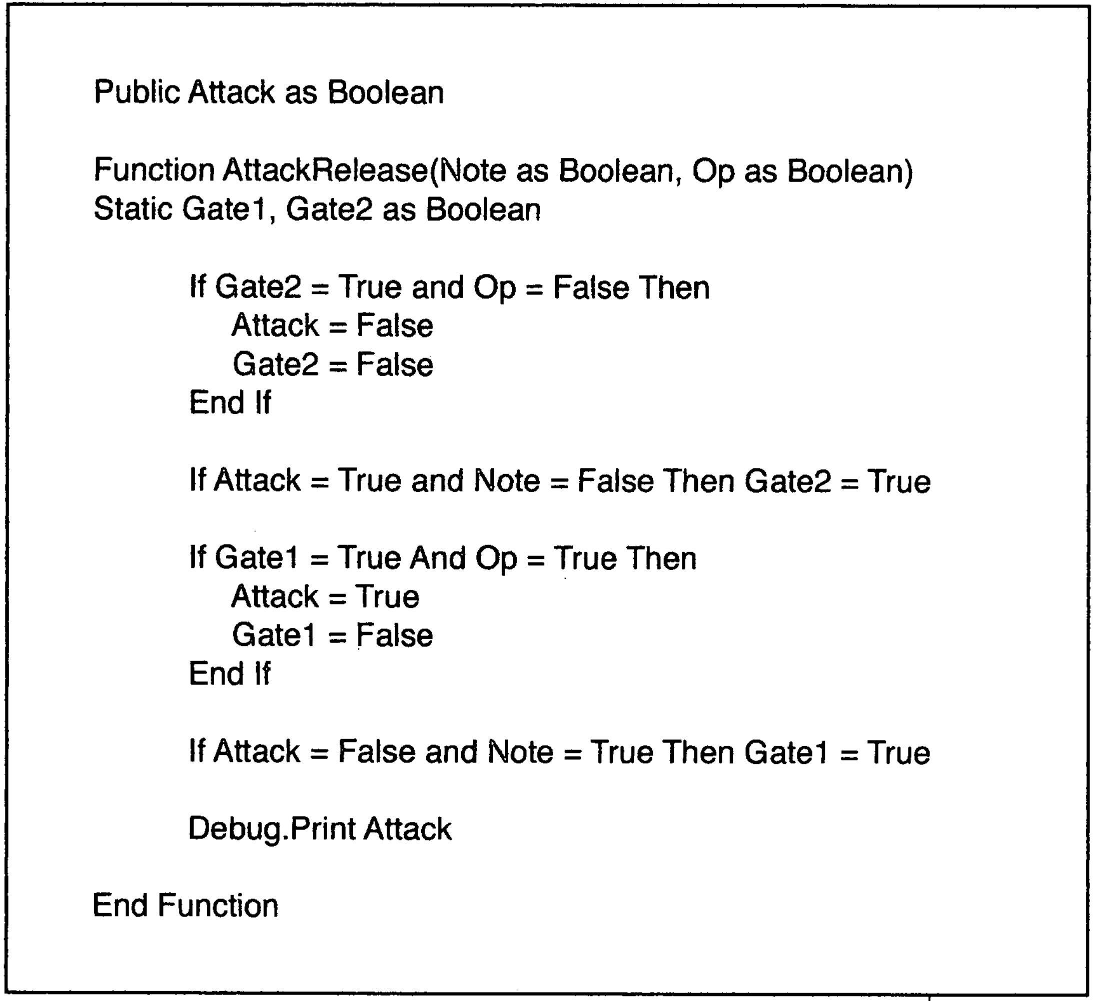
```
Public AttackLeft, AttackRight as Boolean

Function AttRel6(Note As Boolean, Op As Boolean)
Static Gate1, Gate2 as Boolean

    If Note And Op And Gate2 Then
        AttackLeft = True
    End If
    If Note And Not Op And Gate1 Then
        AttackRight = True
    End If
    If Note And Op And Not AttackLeft Then
        Gate1 = True
    End If
    If Note And Not Op And Not AttackRight Then
        Gate2 = True
    End If
    If Not Note And Not Op And Not Gate2 Then
        AttackLeft = False
        Gate1 = False
    End If
    If Not Note And Op And Not Gate1 Then
        AttackRight = False
        Gate2 = False
    End If
    Debug.Print AttackLeft
    Debug.Print AttackRight

End Function
```

FIGURE 17



```
Public Attack as Boolean

Function AttackRelease(Note as Boolean, Op as Boolean)
Static Gate1, Gate2 as Boolean

        If Gate2 = True and Op = False Then
           Attack = False
           Gate2 = False
        End If

        If Attack = True and Note = False Then Gate2 = True

        If Gate1 = True And Op = True Then
           Attack = True
           Gate1 = False
        End If

        If Attack = False and Note = True Then Gate1 = True

        Debug.Print Attack

End Function
```

FORWARD ATTACK LATCH 310

FIGURE 18

```
Public Function AttackD(Note As Boolean, Op As Boolean) As Boolean
Static Gate1, Gate2, Gate3 as Boolean

        If Gate3 And Op Then
           AttackD = True
           Gate3 = False
        End If

        If Gate2 And Note Then
           Gate3 = True
           Gate2 = False
        End If

        If Gate1 And Not Op Then
           AttackD = False
           Gate2 = True
           Gate1 = False
        End If

        If Not Note And Op Then
           Gate1 = True
        End If

        If Not Note And Not Op Then
           Gate2 = True
        End If

End Function
```

FORWARD ATTACK LATCH 310

FIGURE 19

```
Public AttackLeft, AttackRight as Boolean

Function AttackOn(Note As Boolean, Op As Boolean)
Static  Gate1, Gate2, Gate3, Gate4 as Boolean

        If Gate4 = True And Op = False Then
          AttackRight = False
          Gate4 = False
        End If

        If Gate3 = True And Op = False Then
          AttackLeft = False
          Gate3 = False
        End If

        If AttackRight = True And Note = False Then
          Gate4 = True
        End If

        If AttackLeft = True And Note = False Then
          Gate3 = True
        End If

        If Gate2 = True And Op = False Then
          AttackRight = True
          Gate1 = False
          Gate2 = False
        End If

        If Gate1 = True And Op = True Then
           AttackLeft = True
          Gate1 = False
          Gate2 = False
        End If

        If AttackRight = False And AttackLeft = False And Note = True Then
          Gate1 = True
          Gate2 = True
        End If

        If AttackLeft = False And AttackRight = False And Note = True Then
          Gate1 = True
          Gate2 = True
        End If

        Debug.Print AttackLeft
        Debug.Print AttackRight

End Function
```

FIGURE 20

```
Public AttackLeft, AttackRight as Boolean

Function BiAttackD(Note As Boolean, Op As Boolean)
Static Gate1, Gate2, Gate3, Gate4  as Boolean

        If Gate3 And Op Then
            AttackLeft = True
            Gate3 = False
        End If

        If Gate4 And Not Op Then
            AttackRight = True
            Gate4 = False
        End If

        If Gate2 And Op Then
            AttackRight = False
            Gate2 = False
            Gate1 = True
        End If

        If Gate1 And Not Op Then
            AttackLeft = False
            Gate2 = True
            Gate1 = False
        End If

        If Gate2 And Note Then
            Gate3 = True
            Gate2 = False
        End If

        If Gate1 And Note Then
            Gate4 = True
            Gate1 = False
        End If

        If Not Note And Op Then
            Gate1 = True
            Gate4 = False
        End If

        If Not Note And Not Op Then
            Gate2 = True
            Gate3 = False
        End If

        Debug.Print AttackLeft
        Debug.Print AttackRight

End Function
```

FIGURE 21A

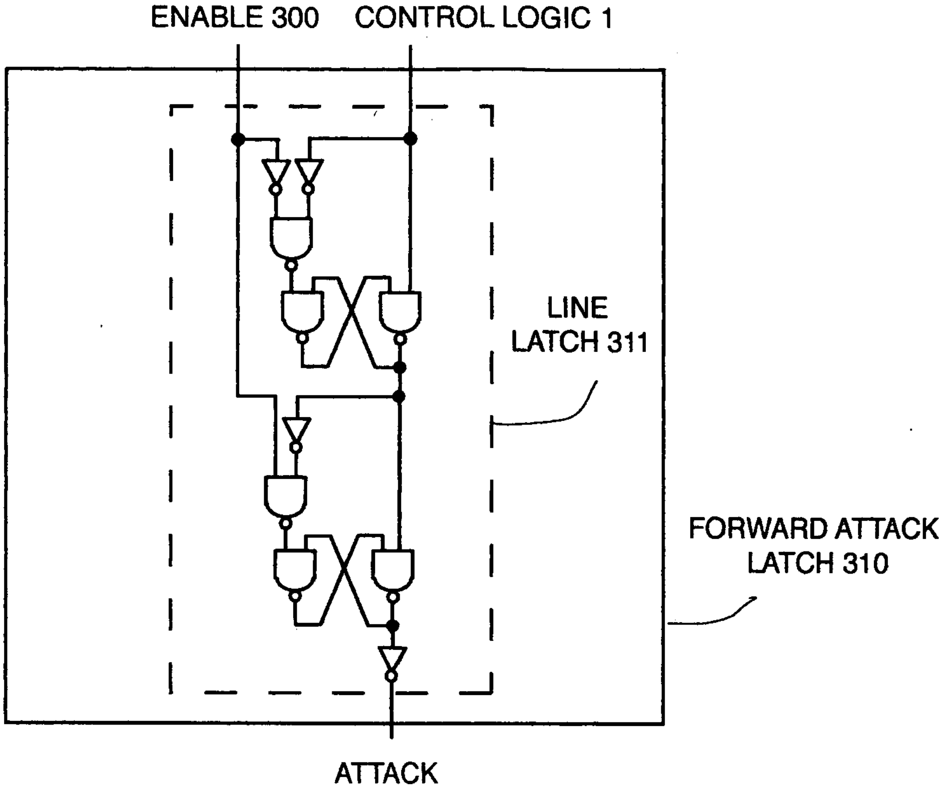


FIGURE 21B

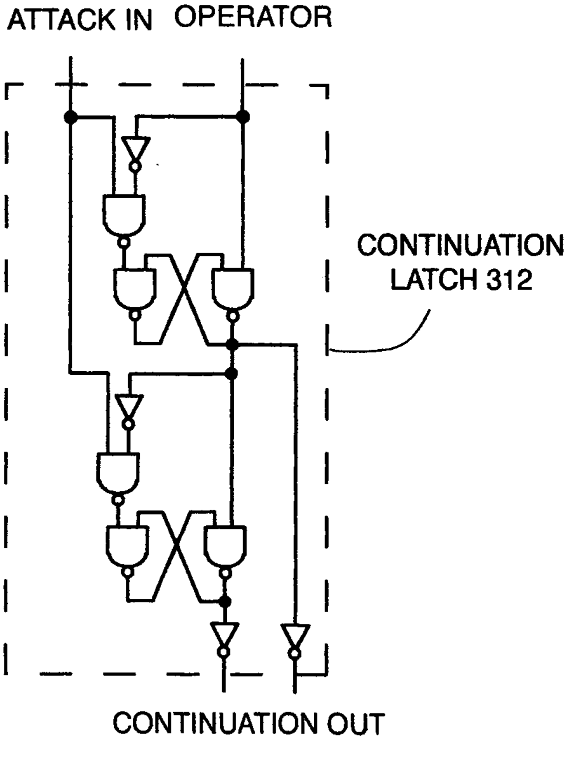


FIGURE 22A

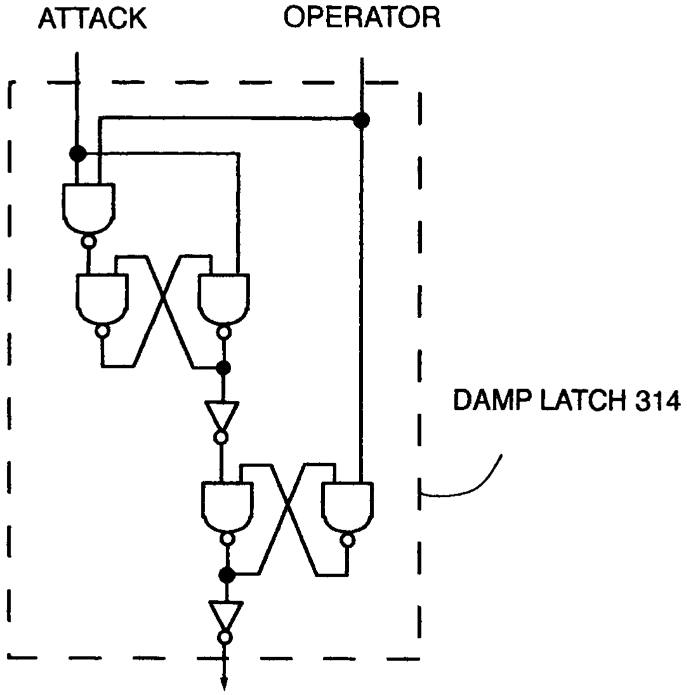


FIGURE 22B

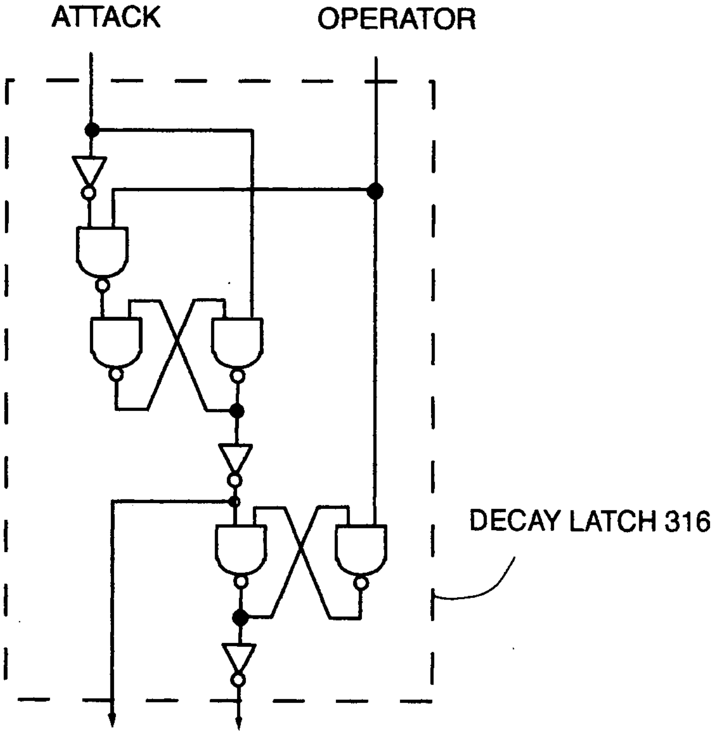


FIGURE 23A

```
Public Note, Op, Attack As Boolean
Public Count As Integer

Function CountLatch(Note, Op)

If Note = 0 And Op = 0 And Count = 0 Then Count = 1
If Note = 1 And Op = 0 And Count = 1 Then
   Count = 2
   Attack = 1
End If

If Note = 0 And Op = 0 And Count = 2 Then Count = 1
If Note = 0 And Op = 1 And Count = 1 Then
   Count = 0
   Attack = 0
End If

Debug.Print Attack
End Function
```

FIGURE 23B

```
Public NoteLast, OpLast, Attack As Boolean
Public NoteTime, OpTime As Variant

Function TimeLatch(Note As Boolean, Op As Boolean)

If Note <> NoteLast Then NoteTime = Time()
If Op <> OpLast Then OpTime = Time()

If Note And Op Then
   If OpTime > NoteTime Then Attack = True
End If

If Not Note And Not Op Then
   If OpTime > NoteTime Then Attack = False
End If

NoteLast = Note
OpLast = Op

Debug.Print Attack
End Function
```

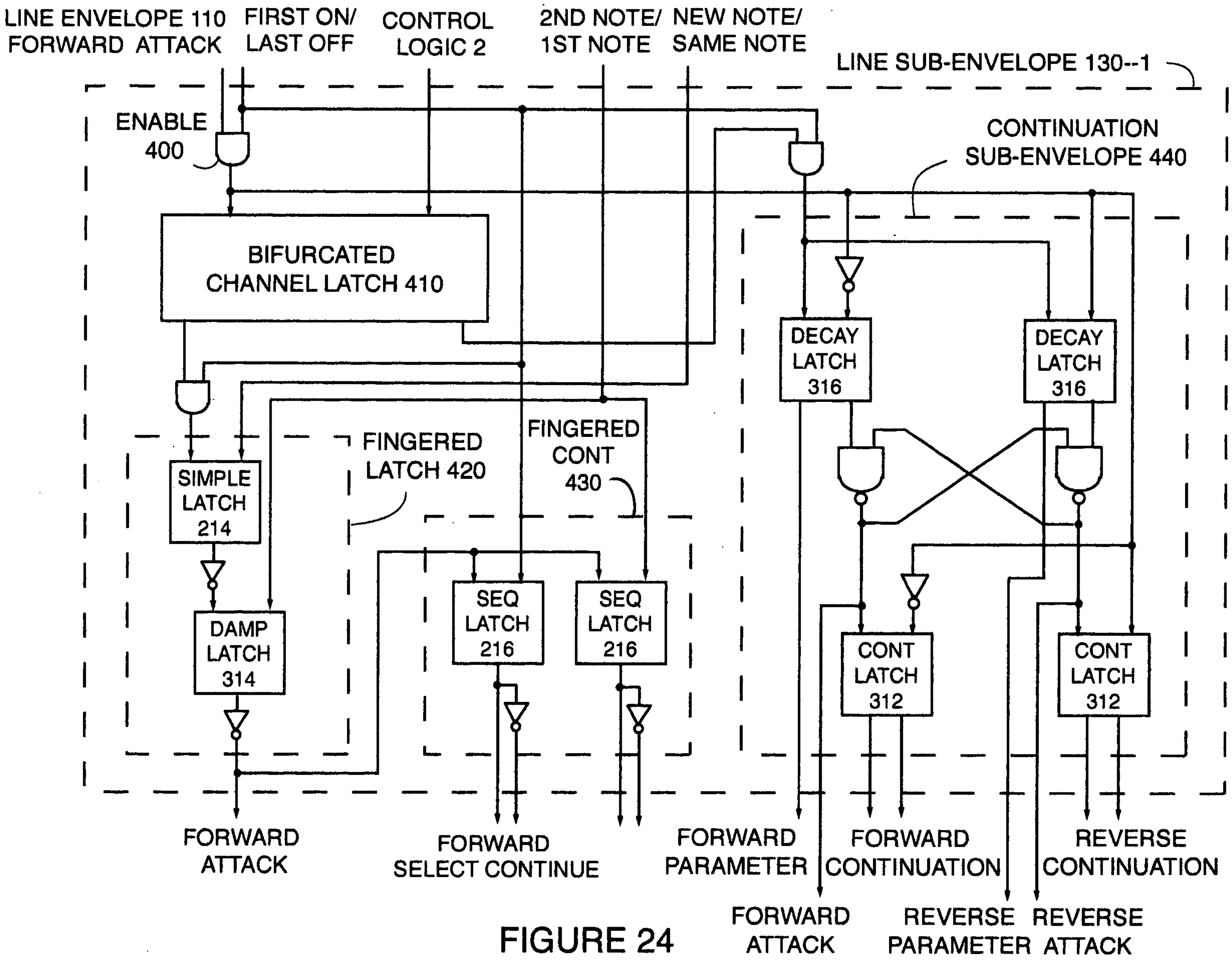

LINE ENVELOPE 110
FORWARD ATTACK
FIRST ON/
LAST OFF
CONTROL
LOGIC 2
2ND NOTE/
1ST NOTE
NEW NOTE/
SAME NOTE
LINE SUB-ENVELOPE 130--1
ENABLE
400
CONTINUATION
SUB-ENVELOPE 440
BIFURCATED
CHANNEL LATCH 410
DECAY
LATCH
316
DECAY
LATCH
316
FINGERED
LATCH 420
FINGERED
CONT
430
SIMPLE
LATCH
214
DAMP
LATCH
314
SEQ
LATCH
216
SEQ
LATCH
216
CONT
LATCH
312
CONT
LATCH
312
FORWARD
ATTACK
FORWARD
SELECT CONTINUE
FORWARD
PARAMETER
FORWARD
CONTINUATION
REVERSE
CONTINUATION
FORWARD
ATTACK
REVERSE
PARAMETER
REVERSE
ATTACK

FIGURE 24

FIGURE 25

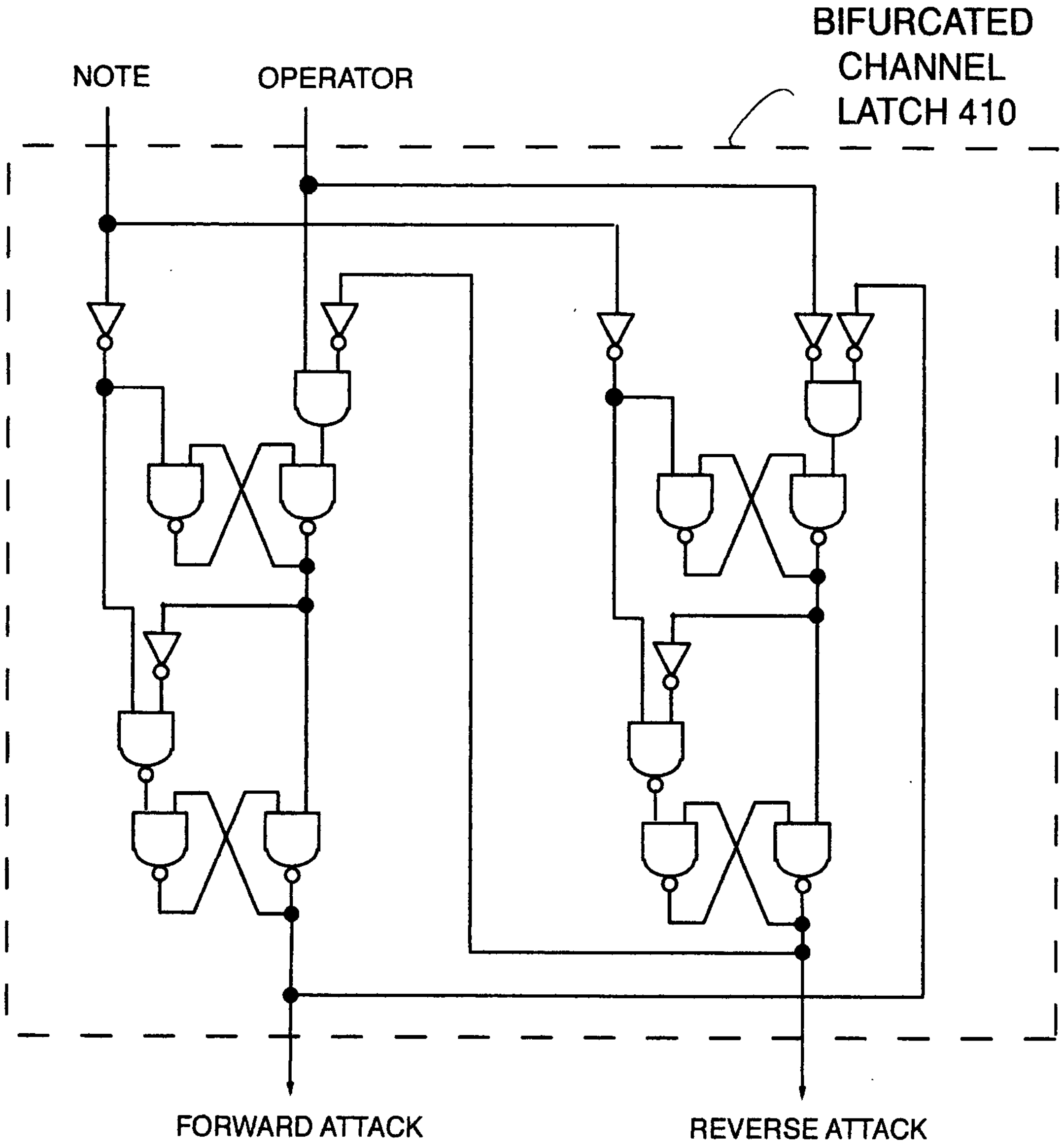
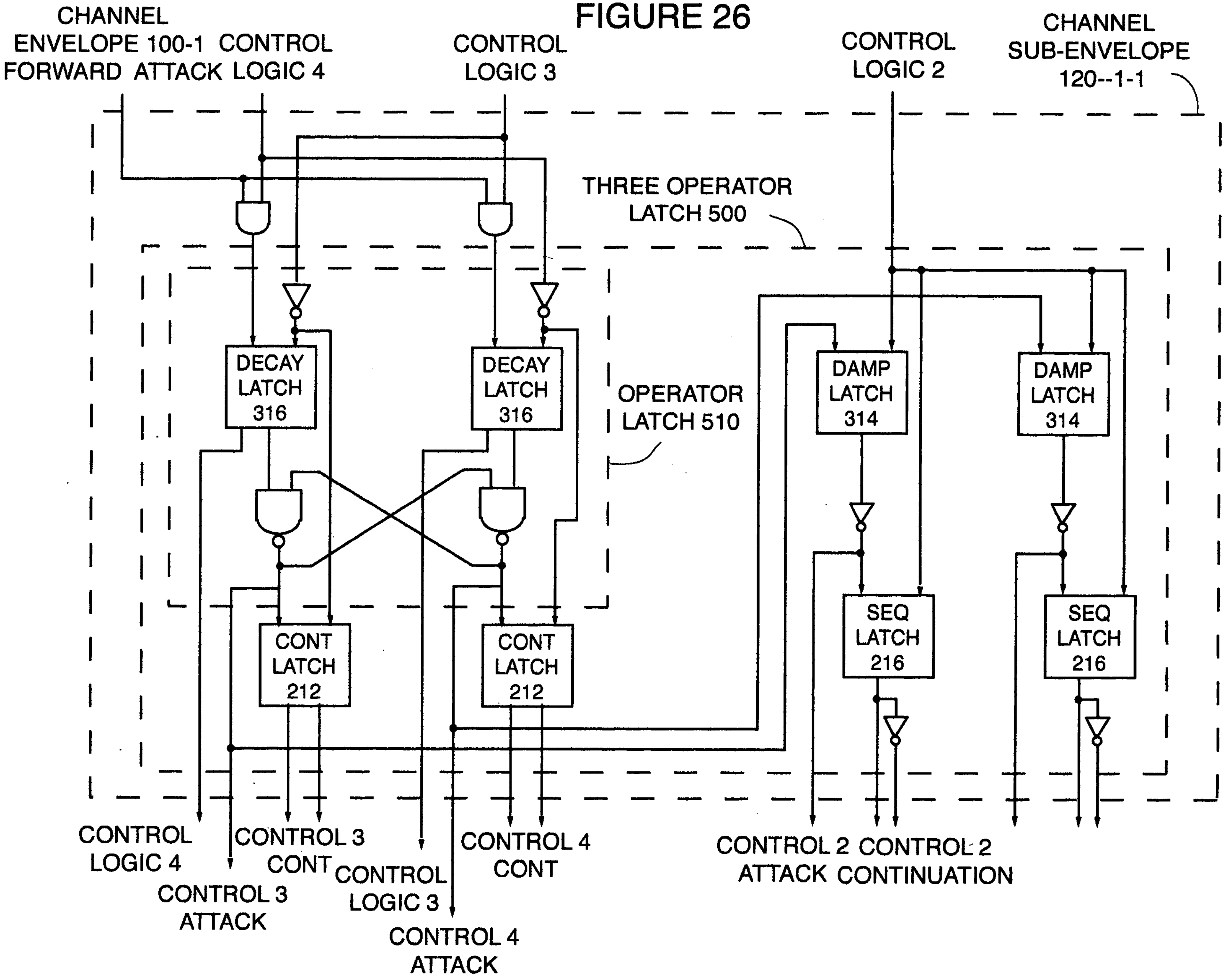

BIFURCATED CHANNEL LATCH 410
NOTE
OPERATOR
FORWARD ATTACK
REVERSE ATTACK

FIGURE 26



CHANNEL ENVELOPE 100-1 FORWARD ATTACK
CONTROL LOGIC 4
CONTROL LOGIC 3
CONTROL LOGIC 2
CHANNEL SUB-ENVELOPE 120--1-1
THREE OPERATOR LATCH 500
OPERATOR LATCH 510
DECAY LATCH 316
DECAY LATCH 316
DAMP LATCH 314
DAMP LATCH 314
CONT LATCH 212
CONT LATCH 212
SEQ LATCH 216
SEQ LATCH 216
CONTROL LOGIC 4
CONTROL 3 ATTACK
CONTROL 3 CONT
CONTROL LOGIC 3
CONTROL 4 ATTACK
CONTROL 4 CONT
CONTROL 2 ATTACK
CONTROL 2 CONTINUATION

FIGURE 27

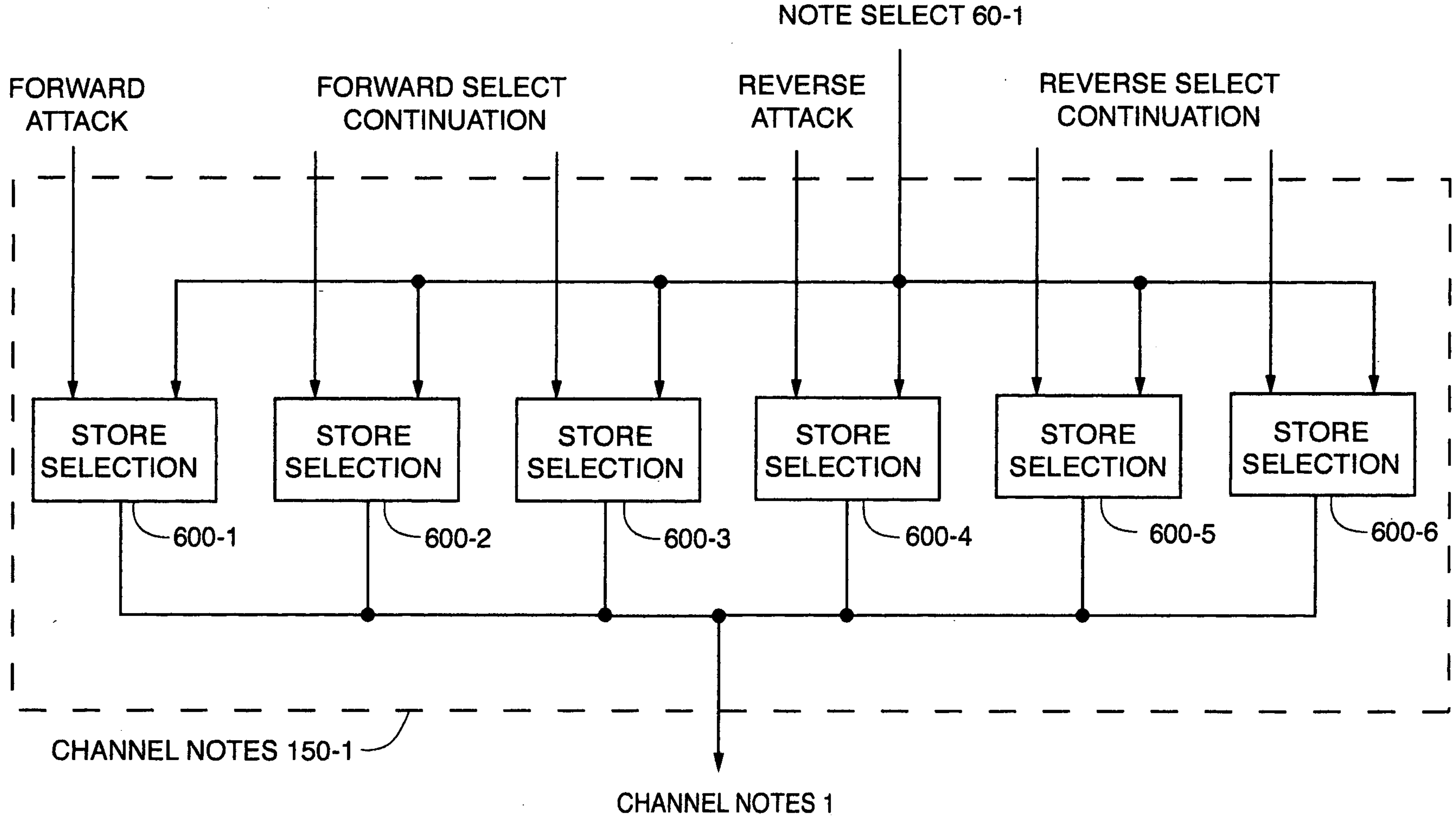

NOTE SELECT 60-1
FORWARD ATTACK
FORWARD SELECT CONTINUATION
REVERSE ATTACK
REVERSE SELECT CONTINUATION
STORE SELECTION
STORE SELECTION
STORE SELECTION
STORE SELECTION
STORE SELECTION
STORE SELECTION
600-1
600-2
600-3
600-4
600-5
600-6
CHANNEL NOTES 150-1
CHANNEL NOTES 1

FIGURE 28

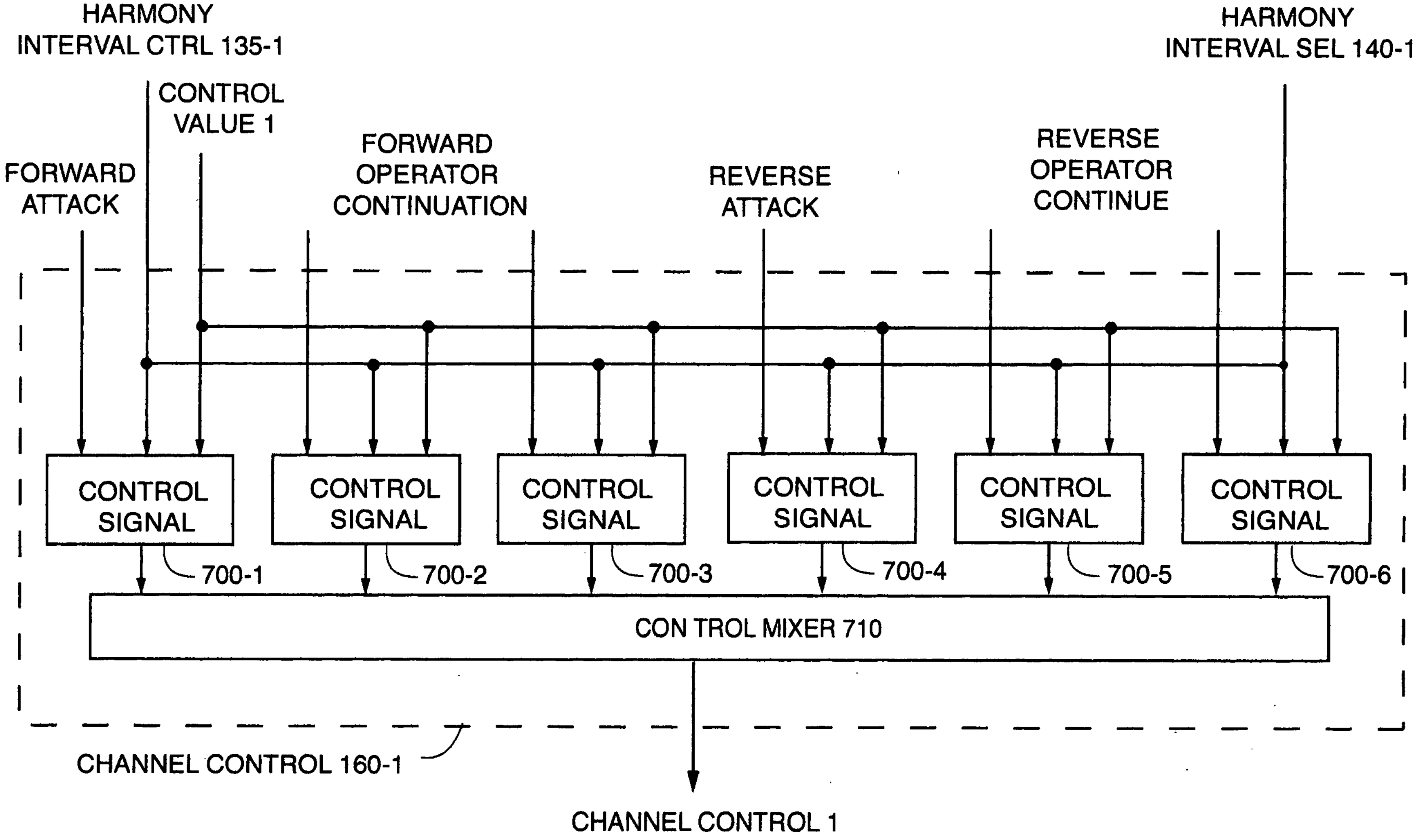

HARMONY
INTERVAL CTRL 135-1
CONTROL
VALUE 1
FORWARD
ATTACK
FORWARD
OPERATOR
CONTINUATION
REVERSE
ATTACK
REVERSE
OPERATOR
CONTINUE
HARMONY
INTERVAL SEL 140-1
CONTROL
SIGNAL
700-1
CONTROL
SIGNAL
700-2
CONTROL
SIGNAL
700-3
CONTROL
SIGNAL
700-4
CONTROL
SIGNAL
700-5
CONTROL
SIGNAL
700-6
CON TROL MIXER 710
CHANNEL CONTROL 160-1
CHANNEL CONTROL 1

FIGURE 29

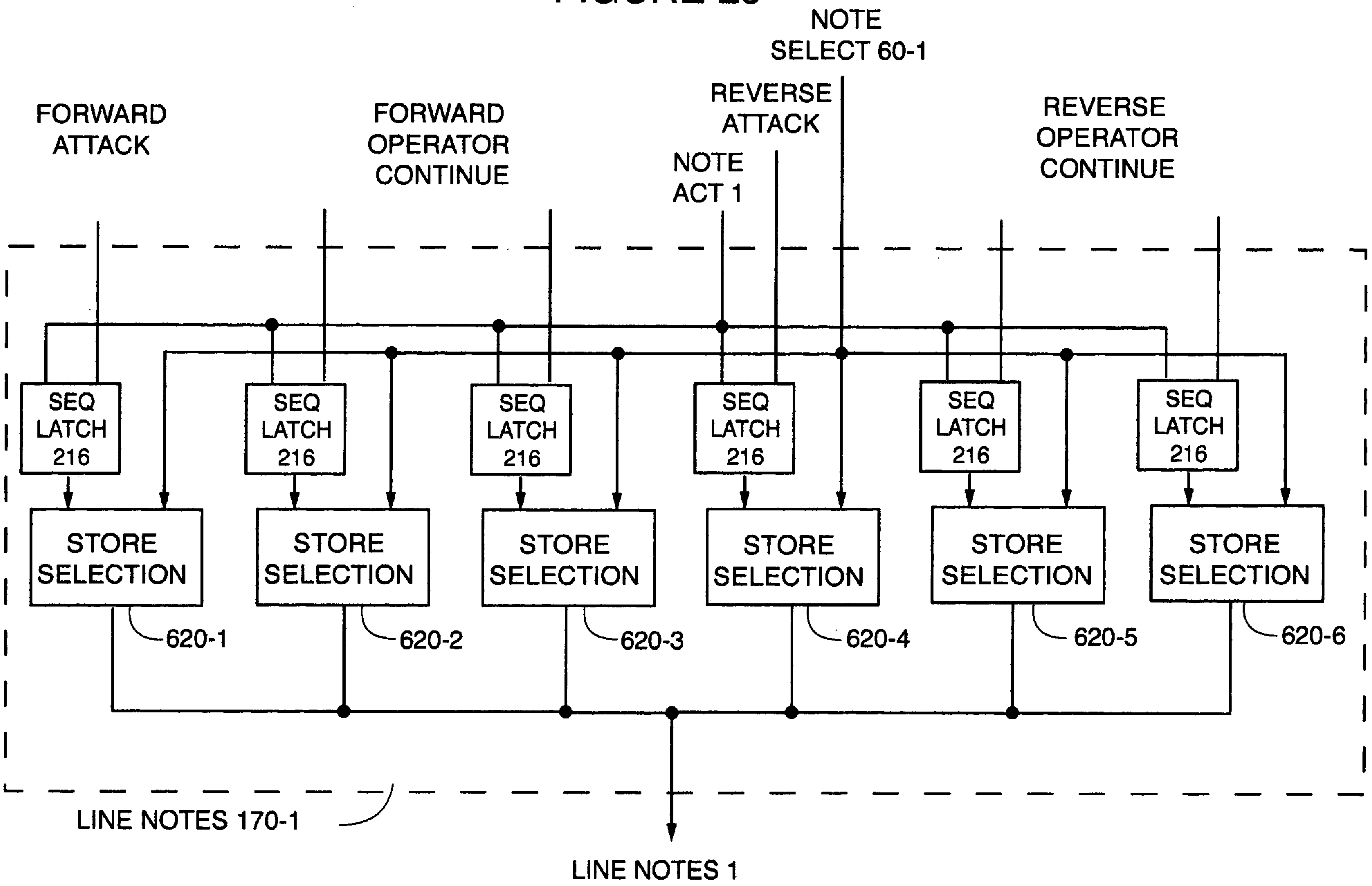

NOTE
SELECT 60-1
FORWARD
ATTACK
FORWARD
OPERATOR
CONTINUE
REVERSE
ATTACK
NOTE
ACT 1
REVERSE
OPERATOR
CONTINUE
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
STORE
SELECTION
STORE
SELECTION
STORE
SELECTION
STORE
SELECTION
STORE
SELECTION
STORE
SELECTION
620-1
620-2
620-3
620-4
620-5
620-6
LINE NOTES 170-1
LINE NOTES 1

FIGURE 30

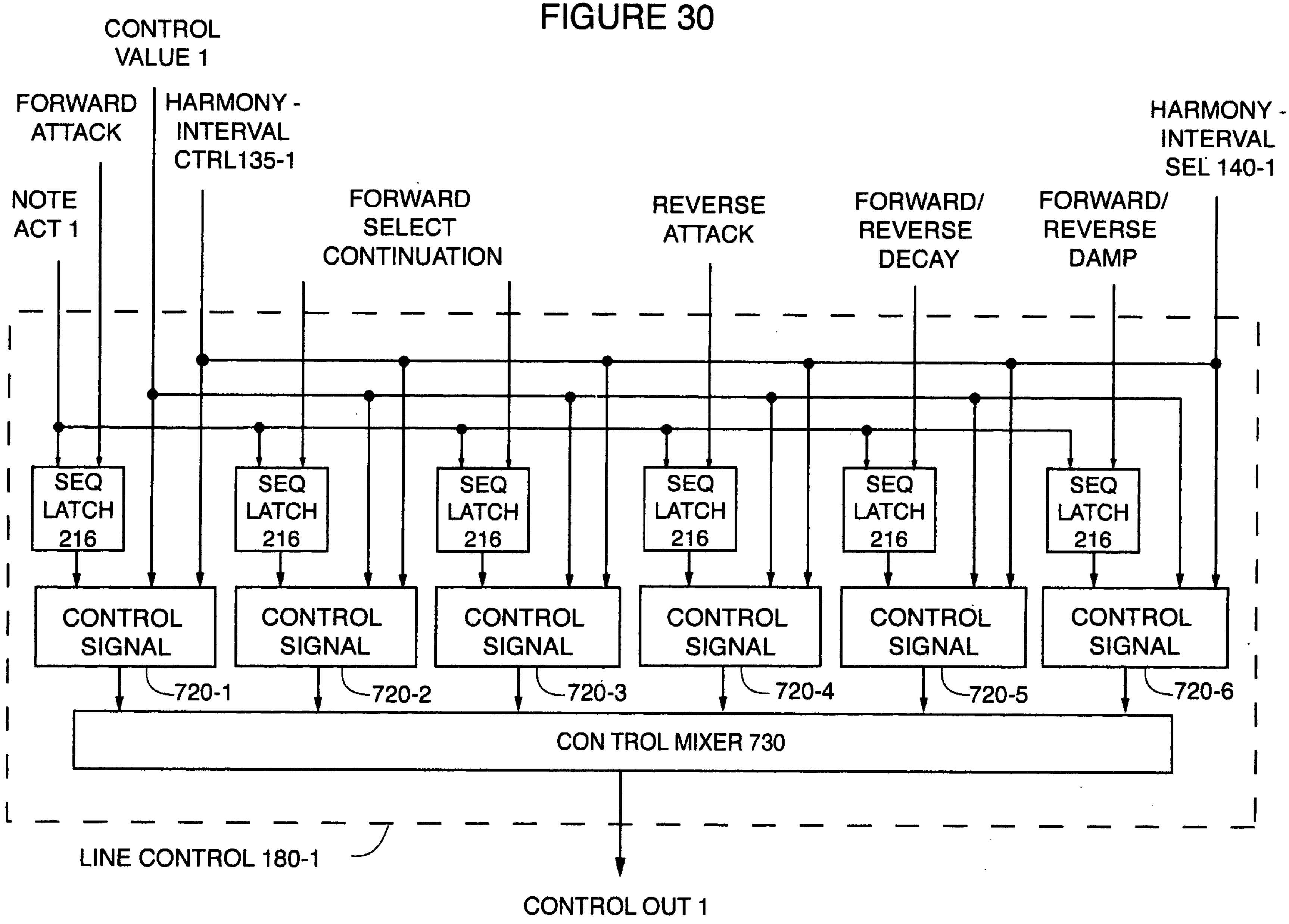

CONTROL
VALUE 1
FORWARD
ATTACK
HARMONY -
INTERVAL
CTRL135-1
NOTE
ACT 1
FORWARD
SELECT
CONTINUATION
REVERSE
ATTACK
FORWARD/
REVERSE
DECAY
FORWARD/
REVERSE
DAMP
HARMONY -
INTERVAL
SEL 140-1
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
SEQ
LATCH
216
CONTROL
SIGNAL
CONTROL
SIGNAL
CONTROL
SIGNAL
CONTROL
SIGNAL
CONTROL
SIGNAL
CONTROL
SIGNAL
720-1
720-2
720-3
720-4
720-5
720-6
CON TROL MIXER 730
LINE CONTROL 180-1
CONTROL OUT 1

FIGURE 31

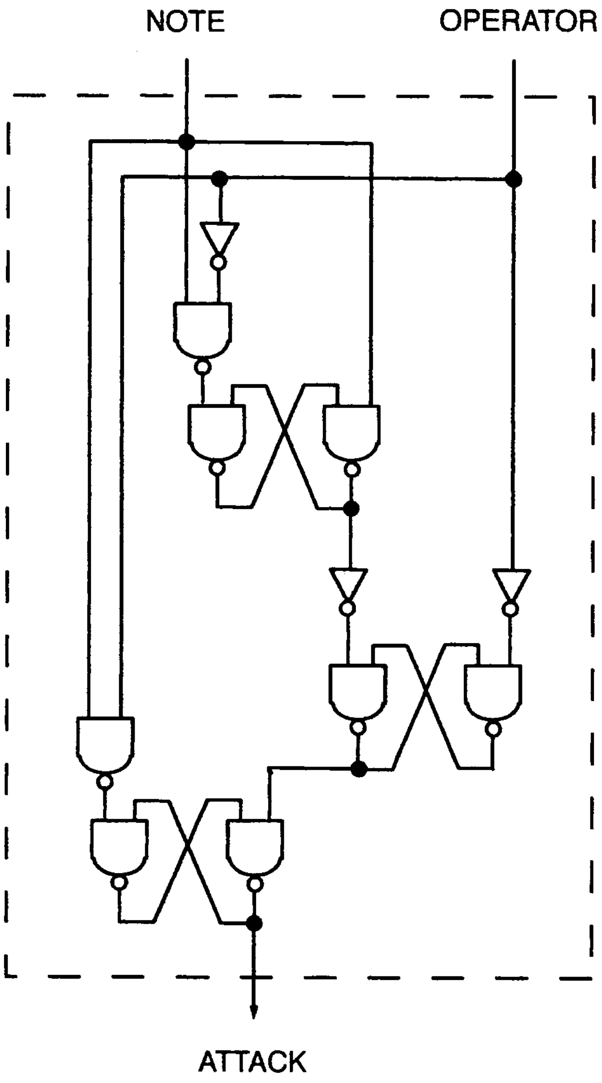


FIGURE 32

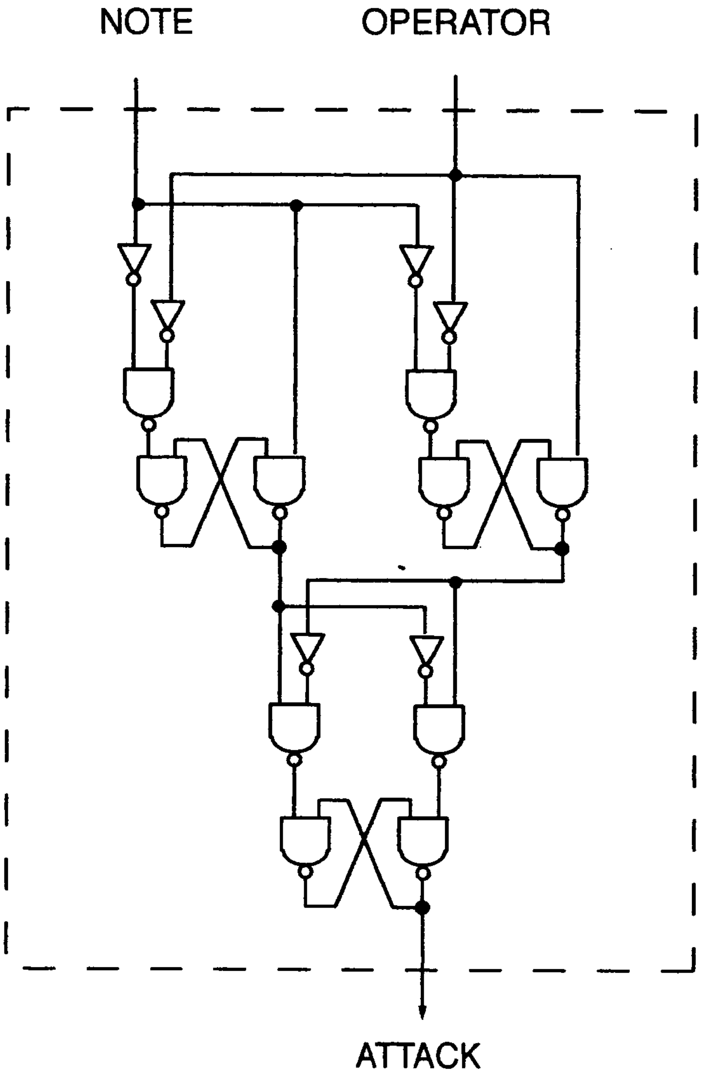


FIGURE 33

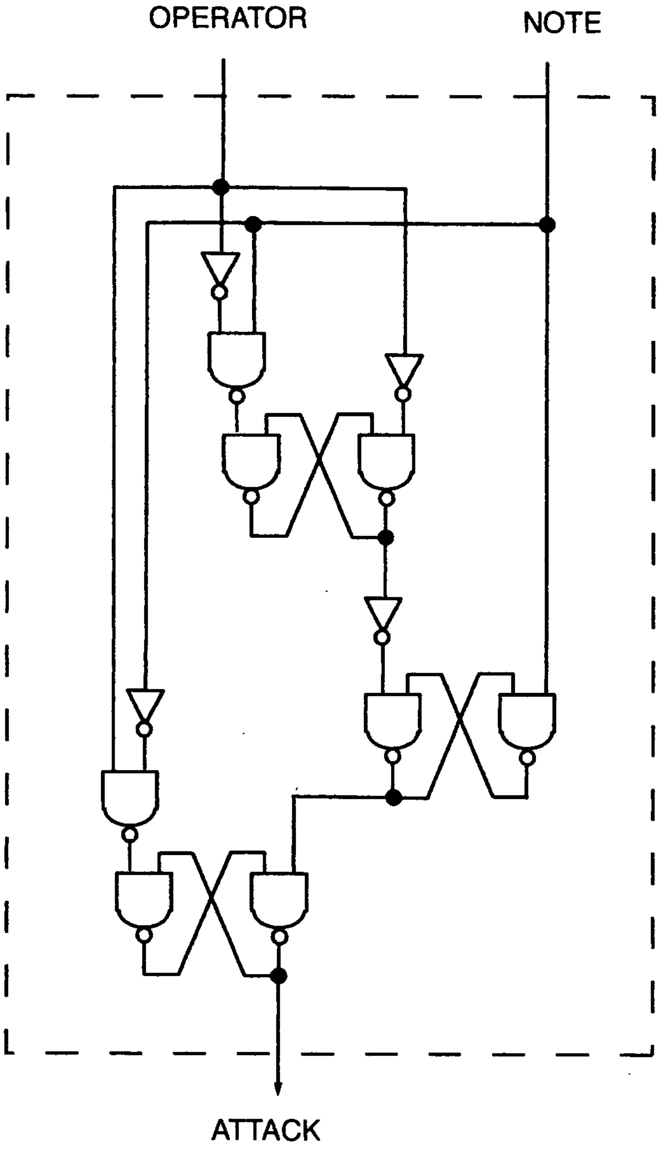


FIGURE 34

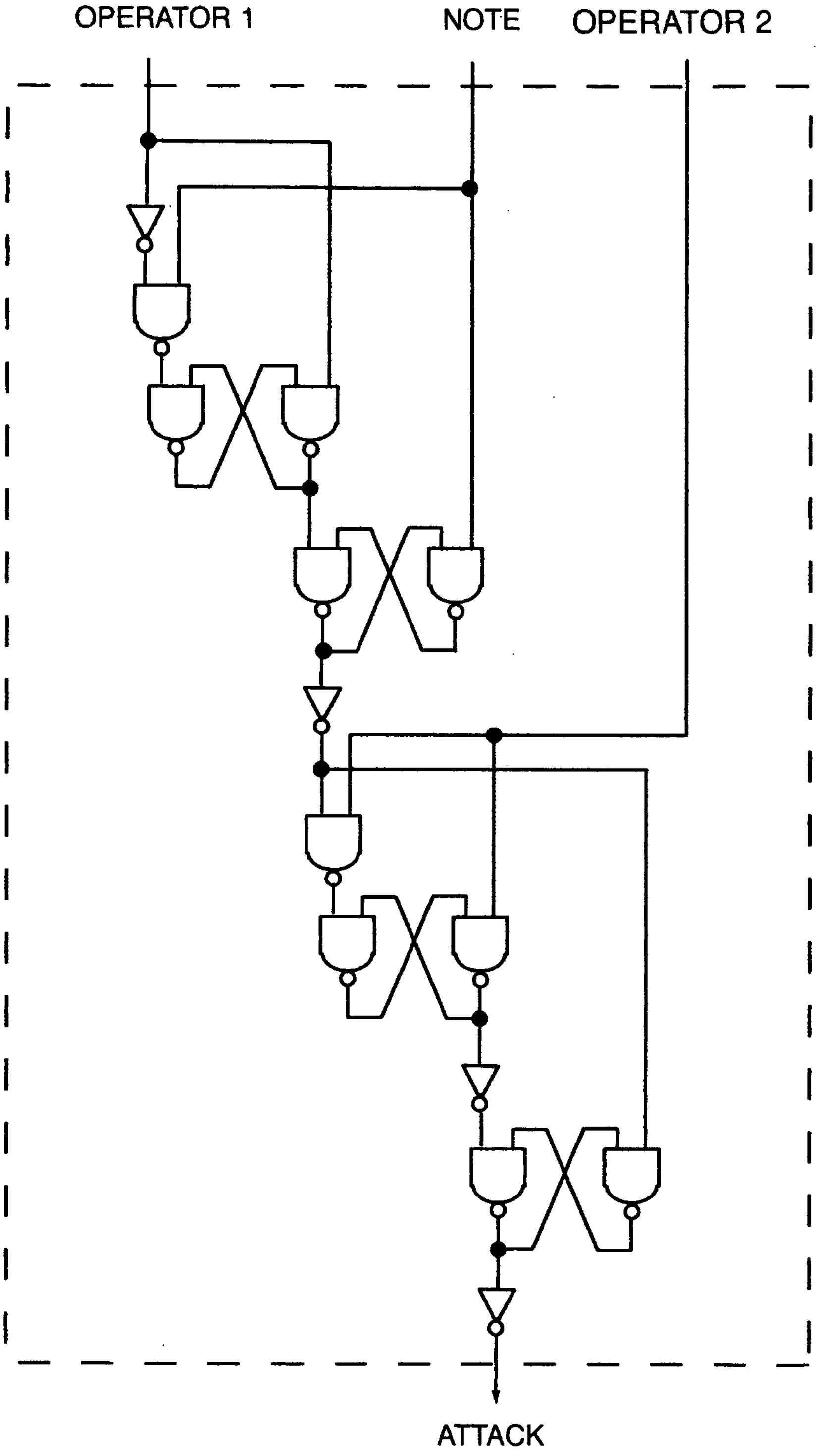
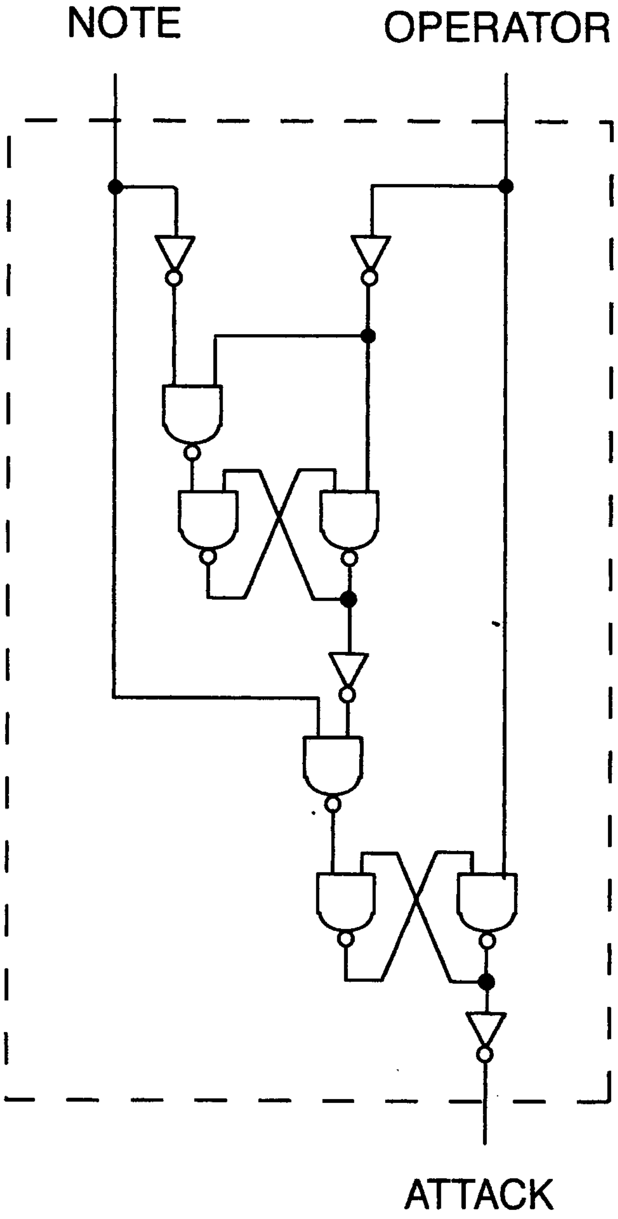


FIGURE 35


NOTE
OPERATOR
ATTACK